

```

using Microsoft.AspNetCore.Builder;
using Microsoft.AspNetCore.Hosting;
using Microsoft.AspNetCore.Identity;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using Microsoft.Extensions.Configuration;
using Microsoft.Extensions.DependencyInjection;
using MyCompany.Domain;
using MyCompany.Domain.Repositories.Abstract;
using MyCompany.Domain.Repositories.EntityFramework;
using MyCompany.Service;

namespace MyCompany
{
    public class Startup
    {
        public IConfiguration Configuration { get; }

        public Startup(IConfiguration configuration) => Configuration = configuration;

        public void ConfigureServices(IServiceCollection services)
        {
            //İâëþ÷ààî êîîðëä èç appsetting.json
            Configuration.Bind("Project", new Config());

            //İâëþ÷ààî íóæíúé ôóíëëííäë ïðëëíæáíëý â èà÷ãñòää ñäðäëñîâ
            services.AddTransient<ITextFieldsRepository, EFTextFieldsRepository>();
            services.AddTransient<IServiceItemsRepository, EFServiceItemsRepository>();
            services.AddTransient<DataManager>();

            //İâëþ÷ààî êîîðäêñò ÄÄ
            services.AddDbContext<AppDbContext>(x =>
x.UseSqlServer(Config.ConnectionString));

            //íàñòðàëëääààî identity ñèñòåì
            services.AddIdentity<IdentityUser, IdentityRole>(opts =>
            {
                opts.User.RequireUniqueEmail = true;
                opts.Password.RequiredLength = 6;
                opts.Password.RequireNonAlphanumeric = false;
                opts.Password.RequireLowercase = false;
                opts.Password.RequireUppercase = false;
                opts.Password.RequireDigit = false;
            }).AddEntityFrameworkStores<AppDbContext>().AddDefaultTokenProviders();

            //íàñòðàëëääààî authentication cookie
            services.ConfigureApplicationCookie(options =>
            {

```

```

options.Cookie.Name = "myCompanyAuth";
options.Cookie.HttpOnly = true;
options.LoginPath = "/account/login";
options.AccessDeniedPath = "/account/accessdenied";
options.SlidingExpiration = true;
});

//îñòðàèààî ïëèèèéó àâîððèàöèè äëü Admin area
services.AddAuthorization(x =>
{
    x.AddPolicy("AdminArea", policy => { policy.RequireRole("admin"); });
});

//äíààäëüàî ñððàèñ äëü èíîððèèððà è ïðàññààèèè (MVC)
services.AddControllersWithViews(x =>
{
    x.Conventions.Add(new AdminAreaAuthorization("Admin", "AdminArea"));
})
//âñòàäëüàî ñîñòàâèññîñ ñ asp.net core 3.0
    .SetCompatibilityVersion(CompatibilityVersion.Version_3_0).AddSessionStateTem
pDataProvider();
}

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
{
    //!!! ïðîäëè ðàäèññòàöèè middleware ïðîäëè àâè

    //à ïðîññà ðàððàíèè íà àâèí àèàòò èàèè èìíí ïðèè
    if (env.IsDevelopment())
        app.UseDeveloperExceptionPage();

    //ïðèèð-àà ïðàððèè ñòàòè÷åñêèå ðàððàíèè à ïðèèèèè (css, js è ò.ä.)
    app.UseStaticFiles();

    //ïðèèè-àà ñèñòèìà ïððððèè
    app.UseRouting();

    //ïðèèè-àà àòàíèèèèèèè è àâððèàöèè
    app.UseCookiePolicy();
    app.UseAuthentication();
    app.UseAuthorization();

    //ðàèññòàöèèè íà íàèíà ïððððèè (äèñíèè)
    app.UseEndpoints(endpoints =>
    {
        endpoints.MapControllerRoute("admin",
            "{area:exists}/{controller=Home}/{action=Index}/{id?}");
        endpoints.MapControllerRoute("default", "{controller=Home}/{action=Index}/{id?}");
    });
}

```

```
    }  
  }  
};
```